



LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

Design of Complex Systems in the presence of Large Uncertainties: a statistical approach

P.S. Koutsourelakis

August 3, 2007

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Design of Complex Systems in the presence of Large Uncertainties: *a statistical approach*

P.S. Koutsourelakis

July 31, 2007

Contents

| | | |
|----------|--|-----------|
| 1 | INTRODUCTION | 3 |
| 2 | PROBLEM DEFINITION | 4 |
| 3 | METHODOLOGY | 6 |
| 3.1 | Sequential Importance Sampling | 6 |
| 3.2 | Probabilistic Classification | 10 |
| 3.2.1 | Kernelized Probit Regression | 15 |
| 3.2.2 | Iterative Calculation | 17 |
| 4 | NUMERICAL EXAMPLES | 18 |
| 4.1 | Single-degree-of-freedom linear oscillator | 19 |
| 4.2 | Embankment Dam | 24 |
| 4.3 | Designing Random Microstructures - Fracture Modeling | 29 |
| 4.4 | Policy Decisions for Energy Systems | 33 |
| 5 | CONCLUSIONS | 35 |

1 INTRODUCTION

The design or optimization of engineering systems is generally based on several assumptions related to the loading conditions, physical or mechanical properties, environmental effects, initial or boundary conditions etc. The effect of those assumptions to the optimum design or the design finally adopted is generally unknown particularly in large, complex systems. A rational recourse would be to cast the problem in a probabilistic framework which accounts for the various uncertainties but also allows to quantify their effect in the response/behavior/performance of the system. In such a framework the performance function(s) of interest are also random and optimization of the system with respect to the design variables has to be reformulated with respect to statistical properties of these objectives functions (e.g. probability of exceeding certain thresholds).

Analysis tools are usually restricted to elaborate legacy codes which have been developed over a long period of time and are generally well-tested (e.g. Finite Elements). These do not however include any stochastic components and their alteration is impossible or ill-advised. Furthermore as the number of uncertainties and design variables grows, the problem quickly becomes computationally intractable.

The present paper advocates the use of statistical learning in order to perform these tasks for any system of arbitrary complexity as long as a deterministic solver is available. The proposed computational framework consists of two components. Firstly advanced sampling techniques are employed in order to efficiently explore the dependence of the performance with respect to the uncertain and design variables. The proposed algorithm is directly parallelizable and attempts to maximize the amount of information extracted with the least possible number of calls to the deterministic solver. The output of this process is utilized by statistical classification procedures in order to derive the dependence of the performance statistics with respect to the design variables. For that purpose we explore parametric and non-parametric (kernel) probit regression schemes and propose an a priori boosting scheme that can improve the ac-

curacy of the estimators. In all cases a Bayesian framework is adopted that produces robust estimates and can also be utilized to obtain confidence intervals.

For that purpose the present paper advocates a framework that allows for calculating the values of response statistics with respect to design variables (the latter are deterministic variables) and provide global information about the sensitivity of those statistics to the design variables of interest.

2 PROBLEM DEFINITION

We identify two sets of input variables. Firstly, design variables $\mathbf{d} \in D \subseteq \mathbb{R}^{n_d}$. These are deterministic parameters, taking values in pre-defined domain D and with respect to which we want to design our system or evaluate its sensitivity. The second set of variables are the uncertain parameters $\mathbf{x} \in \mathbb{R}^{n_x}$ which are characterized by a joint probability density function (pdf) $p(\mathbf{x} \mid \mathbf{d})$ that in general depends on \mathbf{d} without that being necessary. The methods discussed in the paper are in view of a large vector of uncertain variables (i.e. where n_x is in the hundreds or thousands).

We are generally interested in assessing the effect of \mathbf{x} and \mathbf{d} on a response function $f(\mathbf{x}, \mathbf{d}) : \mathbb{R}^{n_x} \times D \rightarrow \mathbb{R}$. Vector-valued output functions can also be considered as it will be discussed later but we present here the scalar case for clarity. The systems considered are assumed complex enough that f is not known explicitly and is in general a highly nonlinear function of its arguments. We will assume though the existence of a deterministic solver that is able to calculate on demand the value of f for specified \mathbf{x} and \mathbf{d} . In the case of a continuum solid mechanics problem, \mathbf{x} and \mathbf{d} can represent any combination of material properties, loading and initial/boundary conditions, f can be a displacement or stress at a point of interest and the deterministic solver is a Finite Element code. Furthermore we will assume that the cost of each evaluation of f is high and essentially the number of such calls during our analysis dominates the solution time. This is consistent with problems of practical interest where each call

to the deterministic solver implies the solution of a large system of nonlinear and/or transient, integro-differential governing equations.

Due to the dependence on \mathbf{x} , f will also be random with an unknown (a priori) pdf. Therefore it does not make sense to examine the sensitivity of f w.r.t. \mathbf{d} but rather its effect of the statistics of f . An obvious such choice would be the expectation :

$$\mu(\mathbf{d}) = E[f(\mathbf{x}, \mathbf{d})] = \int_{\mathbb{R}^{n_x}} f(\mathbf{x}, \mathbf{d}) p(\mathbf{x} | \mathbf{d}) d\mathbf{x} \quad (1)$$

The latter is solely a function of the design variables as the uncertainties have been integrated out. If f represents cost, then one might for example select the design $\mathbf{d}^* \in D$ that minimizes the expected cost $\mu(\mathbf{d})$. Naturally other statistics can be explored depending on the problem at hand. In the rest of this paper, we discuss the effect on the whole cumulative distribution function (cdf) of f i.e. on probabilities of the form:

$$p_{f_0}(\mathbf{d}) = Pr[f(\mathbf{x}, \mathbf{d}) < f_0] = \int_{\mathbb{R}^{n_x}} 1(f(\mathbf{x}, \mathbf{d}) < f_0) p(\mathbf{x} | \mathbf{d}) d\mathbf{x} \quad (2)$$

) which also depends solely on \mathbf{d} ($1(\cdot)$ is the indicator function. Several thresholds f_0 can be considered and in fact given $p_{f_0}(\mathbf{d})$ for a wide enough range of f_0 , they can be readily used to calculate $\mu(\mathbf{d})$ (Equation (1)) or other statistics of f . The tails p_{f_0} are also useful as they commonly appear in reliability based optimization problems as they represent safety constraints that the adopted design must satisfy. Hence we focus the remainder of the paper on methods that allow for the efficient calculation of $p_{f_0}(\mathbf{d})$ for the design values of interest i.e. $\forall \mathbf{d} \in D$.

A brute force approach would be to define some kind of grid in D and for each each grid point $\mathbf{d}_i \in D$ evaluate the integral of Equation (2) anew using approximate or Monte Carlo based methods. It becomes immediately clear that such an approach will quickly become infeasible as n_d (the dimension of \mathbf{d}) and M_d (the size of the discretization in each dimension of \mathbf{d}) grow. Even if a modest number of calls to the deterministic solver is required for each grid point, the total number will grow exponentially fast. Furthermore such an approach is obviously inefficient as it does

not explore any continuity with respect to \mathbf{d} which would allow one to make a good prediction about $p_{f_0}(\mathbf{d}_1)$ if for example $p_{f_0}(\mathbf{d}_2)$ had been estimated for a $\|\mathbf{d}_1 - \mathbf{d}_2\| < \epsilon$. In addition continuity with respect to the threshold f_i might also exist which should be exploited if inference for multiple threshold levels is of interest. Finally, even if an optimal use of the calls to the deterministic solver is made at each grid point, this pertains only to that design value and it is conceivable that more information about the system could have been extracted with the same number of calls but at different points in D .

In several reliability based optimization problems where $p_{f_0}(\mathbf{d})$ represent failure probabilities that we are naturally interested in minimizing, these can take especially small values (i.e. correspond to very rare events). In the context of Monte Carlo based methods, their estimation requires a significant number of calls to the deterministic solver which can constitute a naive implementation infeasible. In the simplified illustration of Figure 1 for example, one can distinguish that the event of interest can attain a wide range of probability values depending on the design value \mathbf{d} .

The present paper advocates a two-step process that consists of two, by-and-large, independent components. The first is an advanced sampling procedure that populates the joint space of uncertainties \mathbf{x} and design variables \mathbf{d} with points in a manner that the most important regions are identified with the least possible number of calls to the deterministic solver. The second component utilizes the samples generated in a probabilistic classification scheme. Both procedures are explained in detail in the subsequent sections and several examples are also provided.

3 METHODOLOGY

3.1 Sequential Importance Sampling

The goal is to generate samples that could be used in the subsequent step by the classifier. In order to facilitate the sampling process we introduce an artificial density

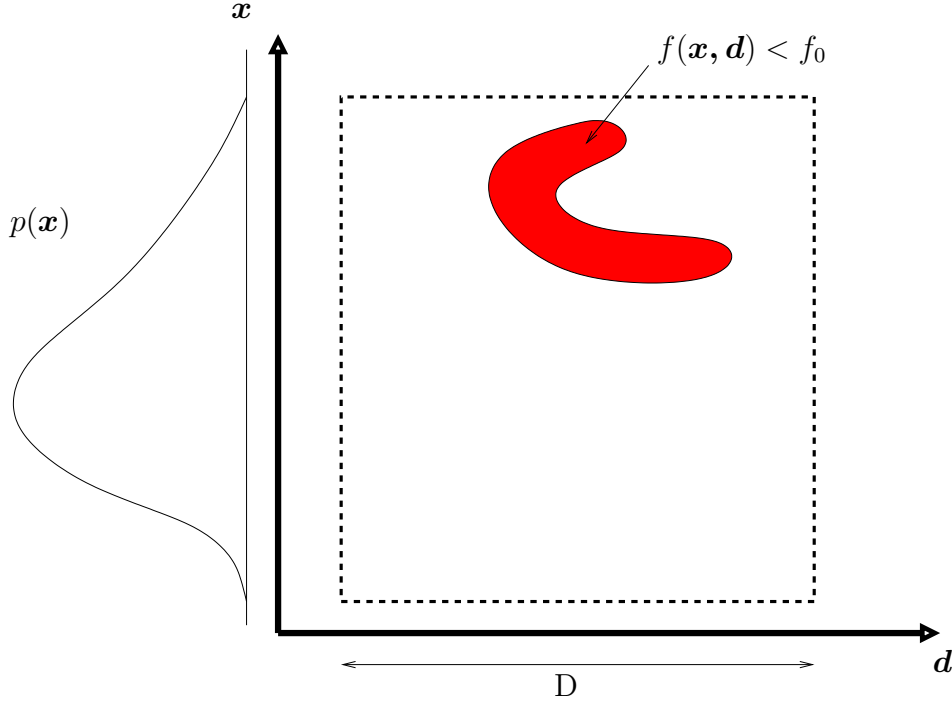


Figure 1: Problem illustration

$\pi_d(\mathbf{d})$ for \mathbf{d} with support on D . This idea was also used successfully in ([2, 7]). It should be noted that the type of distribution does not affect subsequent steps and in the absence of information, the uniform distribution on D would provide the best possible choice. If the analyst has a priori information that would indicate that certain subset(s) of D are more likely to lead to the event of interest (i.e. $f(\mathbf{x}, \mathbf{d}) < f_0$) then a distribution with more power in those regions can be adopted.

Let $\mathbf{y} = (\mathbf{x}, \mathbf{d})$ denote the joint vector and $\pi(\mathbf{y}) = p(\mathbf{x} | \mathbf{d})\pi_d(\mathbf{d})$ the joint pdf. Let f_0 be the threshold of interest and $1_0(\mathbf{y})$ the corresponding indicator function for the event $f(\mathbf{y}) < f_0$. As pointed out earlier, the event of interest can be rare (particularly for small f_0) and simply drawing samples from $\pi(\mathbf{y})$ would generally be inefficient or even infeasible. In order to efficiently explore the parameter space in a manner that requires the least possible number of calls to the deterministic solver (to calculate $f(\mathbf{d})$) and at the same time generate a sufficient number of points for inferring $p_{f_0}(\mathbf{d})$ (see Equation (2)) we employ an iterative importance sampling scheme. The basic concept has been explored in several works ([20, 9, 19, 22]) and represents a combination of

importance sampling and Markov Chain Monte Carlo (MCMC) schemes.

The key ingredient is the introduction of a sequence of distributions $\pi_k(\mathbf{y})$, $k = K, K-1, \dots, 0$ that gradually approximate the target density $\pi_0(\mathbf{y}) = 1_0(\mathbf{y})\pi(\mathbf{y})$ (Note that this is unnormalized but this does not pose any restrictions). In that respect several choices can be adopted. For the problem at hand and in order to utilize the samples generated for infereing the output's f cdf at various other thresholds, it seems that the most natural choice is to define π_k based on a decreasing sequence of thresholds $f_K \geq f_{K-1} \geq \dots f_0$. Thus $\pi_k(\mathbf{y}) \propto 1_k(\mathbf{y})\pi(\mathbf{y})$ where $1_k(\mathbf{y})$ are the indicator functions of the events $f(\mathbf{y}) < f_k$. Trivially we can select $f_K = \infty$ in which case $\pi_K(\mathbf{y})$ coincides with $\pi(\mathbf{y})$.

The basic algorithm consists of three steps, namely Reweighting, Resampling and Rejuvenating ([8]). We start with a population of points $\mathbf{y}_i^{(K)}$, $i = 1, 2, \dots, N$ drawn independently (whenever this is possible) from $\pi_K(\mathbf{y})$. Typically N ranges between 100 and 1000 in the examples presented later. Each sample is associated with a weight $w_i^{(K)} = 1$. Starting with $k = K$ we proceed as follows:

- a) **Reweighting:** Update weights $\hat{w}_i^{(k-1)} = w_i^{(k)} \frac{\pi_{k-1}(\mathbf{y}_i^{(k)})}{\pi_k(\mathbf{y}_i^{(k)})} \quad \forall i = 1, \dots, N$
- b) **Resampling:** Obtain a new population $\hat{\mathbf{y}}_i^{(k-1)}$ by multinomial sampling based on the weights $\hat{w}_i^{(k-1)}$. Set the new associated weights $w_i^{(k-1)} = 1$.
- c) **Rejuvenating:** Obtain the new population $\mathbf{y}_i^{(k-1)}$ by applying a kernel $p(\hat{\mathbf{y}}_i^{(k-1)}, \cdot)$ that leaves π_{k-1} invariant. Set $k = k - 1$ and goto step a).

The goal of the proposed algorithm is to gradually identify and populate with points the region of interest $f(\mathbf{y}) < f_0$, a task that would have been difficult, if not impossible, in one attempt.

From the definition of π_k it becomes apparent that the updated weights $\hat{w}_i^{(k-1)}$ will either be 0 or 1. Hence at the k resampling step only points $\mathbf{y}_i^{(k)}$ for which $f(\mathbf{y}_i^{(k)}) < f_{k-1}$ will be kept and will make up the new population $\hat{\mathbf{y}}_i^{(k-1)}$. This of course would not be true if a different sequence of distributions was adopted. The selection of the

intermediate distributions determines the variance of the weights. The resampling step aims to remove points that have exceptionally small weights and therefore provide little information about the system while promote points which belong in the high probability regions and therefore are most informative. Common practice in sequential importance sampling methods is to resample when the coefficient of variation of the weights exceeds a certain threshold ([12, 18]). For the problem at hand the coefficient of variation of the weights $\hat{w}_i^{(k-1)}$ is given by $\sqrt{\frac{1-p}{pN}}$ where p is the proportion of the $\hat{w}_i^{(k-1)}$ that are equal to 1 (i.e. $p = \frac{1}{N} \sum_{i=1}^N \hat{w}_i^{(k-1)}$). This implies that the intermediate distributions π_k (or equivalently the thresholds f_k) do not have to be specified in advance but can be adaptively determined. In particular, for a pre-defined maximum allowable coefficient of variation, e.g. c_{max} , we can select f_k so that $p = \frac{1}{1+Nc_{max}^2}$, meaning any real number between the pN^{th} and $pN + 1^{th}$ smallest values amongst $f(\mathbf{y}_i^{(k)})$. If $f_k \leq f_0$, then the iterations are stopped. The samples generated can also be used to provide estimates of the 'averaged' value of $p_{f_0}(\mathbf{d})$ under the density $\pi_d(\mathbf{d})$:

$$p_{f_0} = E_{\mathbf{d}}[p_{f_0}(\mathbf{d})] = \int_D \left(\int_{\mathbb{R}^{n_x}} 1(f(\mathbf{x}, \mathbf{d}) < f_0) p(\mathbf{x} | \mathbf{d}) d\mathbf{x} \right) \pi_d(\mathbf{d}) d\mathbf{d} \quad (3)$$

In the aforementioned version of the algorithm where the intermediate distributions/thresholds are determined adaptively based on $p = \frac{1}{1+Nc_{max}^2}$, the latter can be approximated by:

$$p_{f_0} \approx p^{K-1} p_0 \quad (4)$$

where $p_0 = \frac{1}{N} \sum_{i=1}^N \hat{w}_i^{(0)}$. It should be noted that the number of iterations K is determined by the algorithm. Similarly approximations can be obtained for 'averaged' probabilities corresponding to intermediate thresholds $f_k \geq f_0$.

Apart from multinomial resampling, other types can be adopted within the same framework ([11]). The population of points can also increase or decrease at certain iterations ([6]). Furthermore, several possibilities exist for the kernel used in the rejuvenation step. This does not need to be known explicitly and is never used in any computations. Most commonly, and in the absence of better insight a single Metropolis-Hastings (MH) update is applied with a random walk proposal. Naturally several MH

updates can be performed at the expense of course of additional system analyses and calls to the deterministic solver. It is also possible to utilize previous transitions in order to make the rejuvenation step more effective. For example, in [8], it is proposed to use an independent Metropolis-Hastings scheme with a Gaussian proposal with mean and covariance approximated from the current population. It should also be noted that the aforementioned algorithm is highly parallelizable as each point in the population can be rejuvenated separately, i.e. each processor can perform the analysis pertaining to a single point.

In [2] and [7] a method called Subset Simulation ([3]) was used to perform the same task. Although it was derived from a different starting point, it is essentially a special case of the aforementioned algorithm. In [3] it is also discussed how events involving vector-valued output functions f can be accommodated with very simple reformulations. For example if $\mathbf{f}(\mathbf{y}) = (f^{(1)}(y), f^{(2)}(y), \dots, f^{(n)}(y))$ takes values in \mathbb{R}^n and we are interested in sampling for the event $(f^{(1)} < f_0^{(1)}, f^{(2)} < f_0^{(1)}, \dots, f^{(n)} < f_0^{(1)})$ then the algorithm above can still be applied for $f_0 = 0$ and by defining a scalar function $\hat{f}(\mathbf{y}) = \max_n (f^{(1)}(y) - f_0^{(1)}, f^{(2)}(y) - f_0^{(2)}, \dots, f^{(n)}(y) - f_0^{(n)})$.

3.2 Probabilistic Classification

The aforementioned sequential algorithm would produce a number of samples $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{d}_i)$ and their respective function values $f(\mathbf{y}_i)$. In fact for the purposes of determining $p_{f_0}(\mathbf{d})$ (Equation (2)) it suffices to record only the value of the respective indicator function, $1_0(\mathbf{y}_i)$ (which is 1 if $f(\mathbf{y}_i) < f_0$ and 0 otherwise). Two previous approaches that have been followed to perform that task ([2, 7]) make use of Bayes rule in order to express $p_{f_0}(\mathbf{d})$ as:

$$p_{f_0}(\mathbf{d}) = Pr[1_0(\mathbf{x}, \mathbf{d}) = 1 \mid \mathbf{d}] = \frac{p(\mathbf{d} \mid 1_0(\mathbf{x}, \mathbf{d}) = 1) Pr[1_0(\mathbf{x}, \mathbf{d}) = 1]}{p(\mathbf{d})} \quad (5)$$

Obviously $Pr[1_0(\mathbf{x}, \mathbf{d}) = 1] = p_{f_0}$ (given in Equation (3)) which can be estimated using Equation (4)) and $p(\mathbf{d}) \equiv \pi_d(\mathbf{d})$ i.e. the artificial pdf on the design vector \mathbf{d} .

Equation (5) converts the original problem into a density estimation task. The remaining term in Equation (5), $p(\mathbf{d} \mid 1_0(\mathbf{x}, \mathbf{d}) = 1)$ is essentially the conditional pdf of the design variables \mathbf{d} given that the event of interest has occurred i.e. $1_0(\mathbf{x}, \mathbf{d}) = 1$. In [2], this is estimated by constructing a simple histogram of \mathbf{d}_i values of all the samples $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{d}_i)$ for which $1_0(\mathbf{y}_i) = 1$. Although this is straightforward it is generally applicable for \mathbf{d} -spaces of rather small dimension. Histograms are also not known to be robust and generally require a large number of samples (which implies a large computational cost) to alleviate the undesirable effects of binning. Using more elaborate density estimators (i.e. Parzen windows) could alleviate some of these problems but it nevertheless remains a difficult problem.

In [7], the density $p(\mathbf{d} \mid 1_0(\mathbf{x}, \mathbf{d}) = 1)$ is approximated using the Maximum Entropy Principle ([17]). The sufficient statistics are the conditional expectations of \mathbf{d} , i.e. $E[\mathbf{d} \mid 1_0(\mathbf{x}, \mathbf{d}) = 1]$ which are estimated using the samples $\mathbf{y}_i = (\mathbf{x}_i, \mathbf{d}_i)$ for which $1_0(\mathbf{y}_i) = 1$. The approximating density $g(\mathbf{d})$ is of the following form:

$$g(\mathbf{d}) = \frac{1}{Z(\boldsymbol{\lambda})} e^{-\boldsymbol{\lambda}^T \mathbf{d}} \quad (6)$$

where $\boldsymbol{\lambda}$ is the vector of Lagrange multipliers and $Z(\boldsymbol{\lambda})$ is the normalization constant. Under $g(\mathbf{d})$ the design variables are conditionally independent, an assumption that generally will not hold as it is natural to expect that the combined effect of the design variables is not multiplicative. As it is mentioned in [7], second order moments can also be used to improve the approximation. It should be pointed out however that unless the distribution is unimodal several higher order moments must be included. Estimating the latter requires increasingly more samples and therefore additional computational burden. A deficiency that is also common to both of the aforementioned methods is that they do not utilize samples \mathbf{y}_i for which $1_0(\mathbf{y}_i) = 0$ which could provide complementary information. Hence the effect of a region in the design space D that has no samples for which $1_0(\mathbf{y}_i) = 1$ is the same whether $1_0(\mathbf{y}_i) = 0$ or no such samples at all have been drawn in that region.

In this paper we propose a statistical learning technique that utilizes probabilis-

tic classifiers. These are statistical models that are able to predict the class of an item/individual based on a set of covariates which in our case are the design variables \mathbf{d} . It should be noted though that rather than predicting the class we are interested in the probability that a point \mathbf{d} belongs to either class (see Equation (2)). Another possibility that exists is to use $\mathbf{y} = (\mathbf{x}, \mathbf{d})$ as covariates of the classifier which would then serve as a surrogate solver for $1_0(\mathbf{y})$. In practice though such an approach can be problematic as the dimension of \mathbf{x} can easily be in the thousands and a large number of training samples (and consequently calls to the deterministic solver) could be required. Focusing on the design variables \mathbf{d} and on predicting the probability $p_{f_0}(\mathbf{d})$ implies that we have to work in spaces of lower dimension. Furthermore we can improve the accuracy of our predictions by utilizing the iterative process by which the samples were generated in section 3.1.

We utilize the model of probit regression which belongs to the special class of generalized linear models ([10]). Given a point $\mathbf{d}^{(i)}$ and the binary outcome variable $I_i = 1_0(\mathbf{x}, \mathbf{d}_i)$, the model postulates that:

$$Pr[I_i = 1 \mid \mathbf{d}^{(i)}] = \Phi(h(\mathbf{d})) \quad (7)$$

where $\Phi : \mathbb{R} \rightarrow [0, 1]$ is the cumulative distribution function of the standard normal and:

$$h(\mathbf{d}) = a_0 + \mathbf{a}^T \mathbf{d} = a_0 + \sum_k a_k d_k \quad (8)$$

where $\mathbf{a} = \{a_k\}_{k=0}^{n_d}$ are the parameters of the model. Once those have been determined we can readily estimate the probability of interest for any \mathbf{d} . Typically they are determined by maximizing the (log-)likelihood of the labeled training points that were generated in section 3.1. The maximum log-likelihood estimate can be sensitive to the optimization scheme and the number of samples ([15]) For that purpose we adopt a Bayesian formulation for the inference of \mathbf{a} which has the added advantage of providing confidence intervals ([14]). Let $\{\mathbf{d}^{(i)}\}_{i=1}^N$ be the \mathbf{d} -coordinates of the samples generated using the sequential importance sampling scheme and $\{I_i\}_{i=1}^N$ the respective binary

labels (based on the event $f(\mathbf{y}) < f_0$). The likelihood $p(I_i \mid \mathbf{d}_{(i)}, \mathbf{a})$ under the probit model is:

$$p(I_i \mid \mathbf{d}_{(i)}, \mathbf{a}) = \Phi(\mathbf{a}^T \mathbf{d}_{(i)})^{I_i} (1 - \Phi(\mathbf{a}^T \mathbf{d}_{(i)}))^{1-I_i} \quad (9)$$

and for all the N samples $p(\{I_i\} \mid \{\mathbf{d}_{(i)}\}, \mathbf{a}) = \prod_{i=1}^N p(I_i \mid \mathbf{d}_{(i)}, \mathbf{a})$. It becomes immediately apparent that this form is not conjugate to any possible prior on \mathbf{a} and is lacking any structure that could facilitate computation of the posterior by Markov Chain Monte Carlo (MCMC). For that purpose, we follow [1] and introduce the auxiliary variables \mathbf{z}_i which imply the following equivalent model:

$$\begin{aligned} I_i &= \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{if } z_i \leq 0 \end{cases} \\ z_i &= \mathbf{a}^T \mathbf{d}_{(i)} + \epsilon_i \\ \epsilon_i &\sim i.i.d. \ N(0, 1) \\ \mathbf{a} &\sim \pi_a(\mathbf{a}) \end{aligned} \quad (10)$$

where $\pi_a(\mathbf{a})$ is a prior distribution on the vector \mathbf{a} . Hence, given z_i , the I_i become deterministic and the respective likelihood function:

$$p(\{I_i\} \mid \{\mathbf{d}_{(i)}\}, \mathbf{a}) = \prod_{i=1}^N (I_i H(z_i) + (1 - I_i) H(-z_i)) \quad (11)$$

where H is the Heaviside function. Furthermore $p(\{z_i\} \mid \{\mathbf{d}_{(i)}\}, \mathbf{a}) = \prod_{i=1}^N p(z_i \mid \mathbf{d}_{(i)}, \mathbf{a})$ where:

$$p(z_i \mid \mathbf{d}_{(i)}, \mathbf{a}) \propto \exp \left[-\frac{1}{2} (z_i - \mathbf{a}^T \mathbf{d}_{(i)})^2 \right] \quad (12)$$

Given the equations above, the conditional posterior on the model parameters \mathbf{a} is:

$$\begin{aligned} p(\mathbf{a} \mid \{I_i\}, \{\mathbf{d}_{(i)}\}, \{z_i\}) &\propto p(\{I_i\}, \{\mathbf{d}_{(i)}\}, \{z_i\} \mid \mathbf{a}) \pi_a(\mathbf{a}) \\ &\propto p(\{I_i\} \mid \{\mathbf{d}_{(i)}\}, \mathbf{a}) p(\{z_i\} \mid \{\mathbf{d}_{(i)}\}, \mathbf{a}) \pi_a(\mathbf{a}) \\ &\propto \exp \left[-\frac{1}{2} \sum_{i=1}^N (z_i - \mathbf{a}^T \mathbf{d}_{(i)})^2 \right] \pi_a(\mathbf{a}) \end{aligned} \quad (13)$$

In particular, in the case of a normal prior for \mathbf{a} , $\pi_a(\mathbf{a}) = N(\mathbf{0}, \sigma^2 \mathbf{I})$, the conditional posterior $p(\mathbf{a} \mid \{I_i\}, \{\mathbf{d}_{(i)}\}, \{z_i\})$ is still normal, $N(\boldsymbol{\mu}, \mathbf{V})$ where:

$$\begin{aligned}\boldsymbol{\mu} &= \mathbf{V} \mathbf{d} \mathbf{z} \\ \mathbf{V} &= \left(\frac{1}{\sigma^2} \mathbf{I} + \mathbf{d}^T \mathbf{d}\right)^{-1}\end{aligned}\tag{14}$$

where the matrix $\mathbf{d} = (\mathbf{d}_1, \mathbf{d}_2, \dots, \mathbf{d}_N)$. The conditional posterior for each auxiliary variable z_i is a truncated normal:

$$p(z_i \mid I_i, \mathbf{d}_{(i)}, \mathbf{a}) \propto \begin{cases} H(z_i) \exp \left[-\frac{1}{2}(z_i - \mathbf{a}^T \mathbf{d}_{(i)})^2 \right] & \text{if } I_i = 1 \\ H(-z_i) \exp \left[-\frac{1}{2}(z_i - \mathbf{a}^T \mathbf{d}_{(i)})^2 \right] & \text{if } I_i = 0 \end{cases}\tag{15}$$

which is straightforward to sample from.

The reformulation based on Equation (10) offers a convenient framework for MCMC simulation by iteratively applying Gibbs sampling from the conditional posterior densities for \mathbf{a} and \mathbf{z} based on Equations (13), (14) and (15). More elaborate sampling schemes that can result in faster mixing are discussed in [16] but are not utilized in this paper. Given the posterior distribution on \mathbf{a} , we can estimate the expected probability that any new (unobserved) point \mathbf{d}^* will belong in one of the two classes as follows:

$$\begin{aligned}Pr[I(\mathbf{d}^*) = 1 \mid \mathbf{d}^*, \{I_i, \mathbf{d}_{(i)}\}] &= \int Pr[I(\mathbf{d}^*) = 1 \mid \mathbf{d}^*, \{I_i, \mathbf{d}_{(i)}\}, \mathbf{a}] p(\mathbf{a} \mid \{I_i, \mathbf{d}_{(i)}\}) d\mathbf{a} \\ &= \int Pr[I(\mathbf{d}^*) = 1 \mid \mathbf{d}^*, \{I_i, \mathbf{d}_{(i)}\}] p(\mathbf{a} \mid \{I_i, \mathbf{d}_{(i)}\}) d\mathbf{a} \\ &= \int \Phi(\mathbf{a}^T \mathbf{d}^*) p(\mathbf{a} \mid \{I_i, \mathbf{d}_{(i)}\}) d\mathbf{a}\end{aligned}\tag{16}$$

where the density in the integrand is the posterior distribution of \mathbf{a} . Given samples \mathbf{a}_j drawn from this posterior Equation (16) can be approximated as :

$$Pr[I(\mathbf{d}^*) = 1 \mid \mathbf{d}^*, \{I_i, \mathbf{d}_{(i)}\}] \approx \frac{1}{M} \sum_{j=1}^M \Phi(\mathbf{a}_j^T \mathbf{d}^*)\tag{17}$$

These samples can also be readily used to get confidence intervals for $Pr[I(\mathbf{d}^*) = 1 \mid \mathbf{d}^*, \{I_i, \mathbf{d}_{(i)}\}]$.

It is worth pointing out that the parameters a_k provide a quantitative measure of the effect of each design variable d_k on the probability of the event of interest. The

latter is more sensitive to changes in variables with larger (in absolute terms) weights a_k (this assumes that the same scale is used for all d_k). Even though these weights depend on the event of interest, they provide global sensitivity measures. In contrast to partial derivatives of the response (which in the presence of uncertainties are practically meaningless) or of $p_{f_0}(\mathbf{d})$ itself, which provide highly localized information in the neighborhood of the point where they are evaluated, the weights a_k incorporate information about the sensitivity of the statistic of the response across the whole domain D where \mathbf{d} lies. As it will be shown in the numerical examples in the subsequent sections, they can be a very useful and concise indicator for the analyst. It should finally be noted that the relative importance of the design variables d_k can also be quantified using covariate set uncertainty methods ([16]) which are not however explored herein.

3.2.1 Kernelized Probit Regression

As it can be seen from the linear form of the argument of the probit function in Equations (7), (8), for a fixed probability level, the boundary between the two classes is always a hyperplane. Naturally, this assumption will not generally hold even though in certain problems it could provide a good approximation. In order to increase the modeling flexibility, we introduce a non-linear function $\mathbf{G} : \mathbb{R}^{n_d} \rightarrow \Psi$ which maps the original covariates \mathbf{d} to a higher-dimensional feature space Ψ . In this new space we assume that the class separating boundaries are hyperplanes which is equivalent to non-linear surfaces in the original space. This implies that Equation (8) becomes:

$$h(\mathbf{d}) = a_0 + \boldsymbol{\beta}^T \mathbf{G}(\mathbf{d}) \quad (18)$$

In order to avoid working in the high-dimensional feature space Ψ we assume that the parameter vector $\boldsymbol{\beta}$ can be expressed in terms of the training points $\{\mathbf{d}^{(i)}\}_{i=1}^N$ (or a subset thereof), namely ([24]):

$$\boldsymbol{\beta} = \sum_{i=1}^N a_i \mathbf{G}(\mathbf{d}^{(i)}) \quad (19)$$

Substituting in Equation (18) leads to:

$$h(\mathbf{d}) = a_0 + \sum_{i=1}^N a_i (G^T(\mathbf{d}^{(i)})G(\mathbf{d})) \quad (20)$$

The expression above involves only inner products in the feature space and by making use of Mercer's theorem can be substituted by a continuous, symmetric, positive semi-definite kernel function $K(.,.) : \mathbb{R}^{n_d} \times \mathbb{R}^{n_d} \rightarrow \mathbb{R}^+$. Hence:

$$h(\mathbf{d}) = a_0 + \sum_{i=1}^N a_i K(\mathbf{d}^{(i)}, \mathbf{d}) \quad (21)$$

This implies that the mapping to the feature space is never explicitly defined and all the calculations involved can be readily performed using the kernel function. One of the kernel functions that fulfills the aforementioned conditions and will be utilized in the sequence is the Gaussian radial basis function:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\lambda \frac{\|\mathbf{x} - \mathbf{y}\|^2}{2}\right) \quad (22)$$

In practical terms, Equations (21) and (22) imply that the class-separating boundaries is constructed by the weighted influence of the training points. The range of influence of each of those points depends on the parameter λ which in fact does not have to be common for all the points (heteroscedastic). Another interesting possibility, which is not further examined is to assume the following anisotropic Gaussian kernel:

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{1}{2}(\mathbf{x} - \mathbf{y})^T \mathbf{\Gamma} (\mathbf{x} - \mathbf{y})\right) \quad (23)$$

where the $n_d \times n_d$ matrix $\mathbf{\Gamma}$ is of the form $\mathbf{\Gamma} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_d})$, i.e. it has a different spread in each direction. In such a case, the relative values of the λ_i 's provide a quantitative indicator of the sensitivity of the system to the design variables. Namely large λ_i imply that the probability of interest is less sensitive to variations in the d_i design variable.

Inference of the vector \mathbf{a} is generally more burdensome than in traditional regression as the dimension of the weight vector is equal to the number of training points N

rather than n_d . It can nevertheless be carried out in a Bayesian framework using the same MCMC schemes explained earlier that require iterative block Gibbs sampling. It should also be noted that there are techniques that utilize a subset of the training points N (and in particular the ones closer to the class-separating boundary) in Equation (21) and therefore reduce the number of weights a_i that need to be determined. These techniques are not utilized herein but they could potentially improve the overall performance of the algorithm.

3.2.2 Iterative Calculation

As demonstrated in section 3.1, the generation of the training samples can be much more efficiently performed in an iterative manner by defining a sequence of distributions $\pi_k(\mathbf{y}) \propto 1_k(\mathbf{y})\pi(\mathbf{y})$ where $1_k(\mathbf{y})$ are the indicator functions of the events $f(\mathbf{y}) < f_k$ such that $f_0 < f_1 < \dots < f_K = +\infty$. Similarly to the target event we can define binary variables $I^{(k)}$ that correspond to the k^{th} intermediate event. It is obvious that the probability of interest can be expressed as:

$$\begin{aligned} Pr[I^{(0)}(\mathbf{d}) = 1 \mid \mathbf{d}] &= \sum_{j=0}^1 Pr[I^{(0)}(\mathbf{d}) = 1, I^{(1)}(\mathbf{d}) = j \mid \mathbf{d}] \\ &= Pr[I^{(0)}(\mathbf{d}) = 1, I^{(1)}(\mathbf{d}) = 1 \mid \mathbf{d}] \\ &= Pr[I^{(0)}(\mathbf{d}) = 1 \mid I^{(1)}(\mathbf{d}) = 1, \mathbf{d}] Pr[I^{(1)}(\mathbf{d}) = 1 \mid \mathbf{d}] \quad (24) \end{aligned}$$

where the second equality is a result of the nested definition of the intermediate events.

By proceeding in the same manner we can arrive at:

$$Pr[I^{(0)}(\mathbf{d}) = 1 \mid \mathbf{d}] = \prod_{k=0}^{K-1} Pr[I^{(k)}(\mathbf{d}) = 1 \mid I^{(k+1)}(\mathbf{d}) = 1, \mathbf{d}] \quad (25)$$

Each of the terms in the product above can be approximated using the (kernelized) probit regression schemes presented earlier. The training sample for the k^{th} model should consist though of the points such that $I^{(k+1)}(\mathbf{d}) = 1$. Based on the sequential importance sampling algorithm in section 3.1, one should use the points $\mathbf{y}_i^{(k+1)}$ (and in particular their \mathbf{d} -coordinates) as those always satisfy the condition $I^{(k+1)}(\mathbf{d}) = 1$.

For example for $k = K - 1$ above and since $f_K = +\infty$, we use the initial population of points $\mathbf{y}_i^{(K)}$.

In cases where the probability of interest varies significantly in the design domain D , it is difficult to expect that a single probit regression scheme (kernelized or not) will provide a good approximation for all probability levels. Using the product decomposition scheme of Equation (25), each probit regression is fitted on a subset the whole domain and can therefore provide a better local approximation.

4 NUMERICAL EXAMPLES

As mentioned earlier, the method proposed consists of two, largely independent steps. In the first, the combined space of random and design variables is sampled in order to extract information (in the form of samples) about the event of interest. In all the examples presented the adaptive version of the proposed algorithm was used. The samples generated were in turn used for training the probabilistic classifier at the second step. It should be noted that the training samples were always rescaled in the $[0, 1]^{n_d}$ hypercube in order to achieve consistency of the algorithm. The training task is carried out in a Bayesian framework. The following priors were adopted:

- $N(0, 1)$ for the a_k 's (Equation (7)) or a_i 's (Equation (21))
- $\lambda \sim \text{Gamma}(a_\lambda, s a_\lambda)$ and $s \sim \text{Exp}(a_s)$ with $a_\lambda = 1.0$ and $a_s = 1.0$.

It should be noted that the parameter λ of the kernel function has a smoothing effect. Smaller values of λ imply smaller variation of the probability of interest with respect to \mathbf{d} and vice versa. The advantage of the Bayesian scheme is that by appropriately selecting the prior on λ , the analyst can effectively decide on the smoothing of results if such prior information is available. By selecting a hierarchical prior as above, the data have the most profound effect in determining the best fit. In all the results present above 10,000 MCMC iterations were performed and the first 1000 were discarded as

burn-in time. The estimates presented involve the posterior expected values as calculated from Equation (16). Details on the sequential importance sampling algorithm are provided on a case-by-case basis below.

4.1 Single-degree-of-freedom linear oscillator

This is indeed a trivial example with minimal practical importance as the deterministic solver is not complex and does not impose a significant computational burden. This allows however the calculation of reference solutions with a brute-force method that can be used to validate the proposed algorithms. Furthermore we illustrate some algorithmic details and compare the benefits of the two probit regression schemes presented earlier. The governing equation is:

$$\ddot{x} + 2 \xi \omega_0 \dot{x} + \omega_0^2 x = r(t), \quad t \in [0, 10] \quad (26)$$

where ξ is the damping ratio, ω_0 the natural frequency and $r(t)$ the external excitation. We consider a discretized version of the equation above with $\Delta t = 0.01$. We model the discretized load by independent, standard normal variables and examine the response of the system with respect to the event that the absolute displacement $x(t)$ exceeds a specified threshold x_0 at least once, namely $\exists t_k = k\Delta t \in [0, 10] : |x(t_k)| > x_0 = 0.006$. Our goal is to find the dependence of the probability of this event with respect to the natural frequency. This example was also examined in [7] for $\omega_0^2 \in D \equiv [800, 1200]$ and $\xi = 0.05$.

A reference solution was found by performing independent runs of the sequential importance sampling algorithm at 10 equally spaced grid-points in D and is depicted in Figure 2 (red circles). We used $N = 1000$ at each step and the intermediate thresholds were determined adaptively for $c_{max} = 0.1$ (i.e. $p = 0.1$). The low dimensionality of the problem and almost log-linear dependence on the design variable $d = \omega_0^2$ imply that previously developed algorithms in [2] and [7] would also perform quite well as reported in the respective papers. . We used a uniform distribution on D and applied the

sequential importance sampling algorithm with $N = 100$ and $N = 500$ which led to p_{f_0} (Equations (3), 4)) of 5.2×10^{-3} and 3.18×10^{-3} respectively. The algorithm required 3 iterations (as well as 300 and 1500 calls to the deterministic solver respectively) and the samples produced were utilized in order to fit 3 models of the probit regression based on the iterative calculation of subsection 3.2.2. Figure 2 depicts estimates obtained using Equation (25). Despite the very small probability values, and the large number of random variables (1000), it can be readily seen that the proposed method provides very good estimates across the whole range of the design variable. Naturally the quality of the approximation improves as the number of samples used for training grows. The advantage of the proposed method becomes obvious if one compares the number of calls to the deterministic solver required by a brute-force approach. Even if $N = 100$ are used at each step and with 10 grid points this implies on average $100 \times 10 \times 3 \text{ steps} = 3000 \text{ runs}$ whereas our method would require 300 or 1500 calls to the solver.

We also considered the case that the damping ratio ξ is a design variable taking values in the interval $[0.05, 0.10]$. For $x_0 = 0.005$, we used again a uniform distribution on D and applied the sequential importance sampling algorithm with $N = 100$ and $N = 500$ which led to p_{f_0} (Equations (3), 4)) of 6.5×10^{-3} and 5.06×10^{-3} respectively. The algorithm required 3 iterations (as well as 300 and 1500 calls to the deterministic solver respectively) and the samples produced were utilized in order to fit 3 models of the probit regression based on the iterative calculation of subsection 3.2.2. Figure 3 compares the estimates along two slices of the design space with the reference solution calculated by performing independent runs of the sequential importance sampling algorithm at 10 equally spaced grid-points in D (black circles). Also Figure 4 depicts the estimates obtained using 500 samples at each level on the whole design space D .

A question that is commonly posed in engineering problems relates to the sensitivity of the system's response on the various input parameters. As pointed out earlier, the parameters a_k in the probit regression model (Equation (8)) provide a quantitative

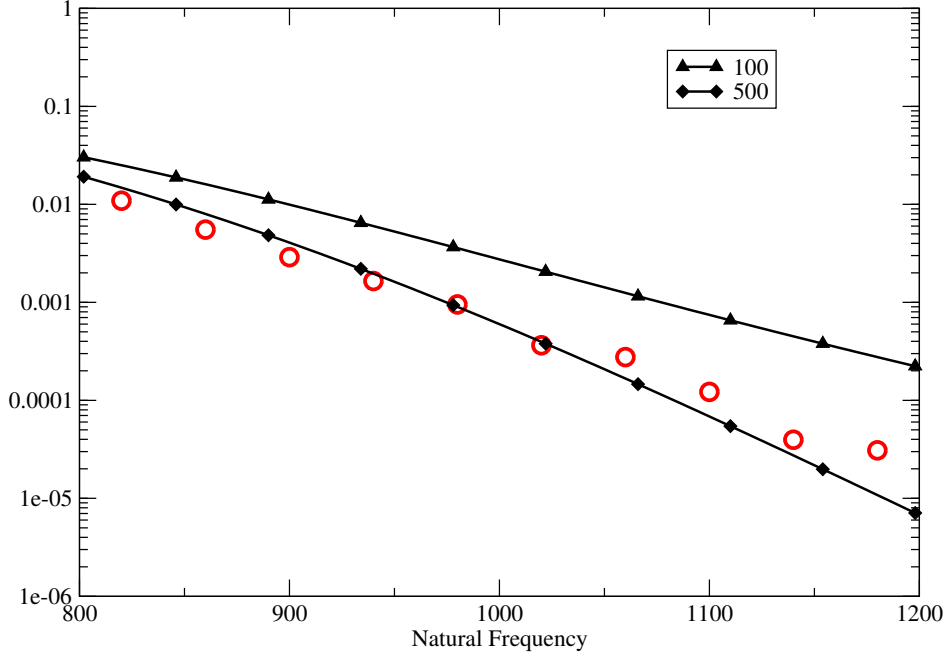


Figure 2: Comparison of reference solution (red circles) with estimates obtained using the probit regression schemes with 3 levels and 100 (triangle up) and 500 (diamond) training samples at each level

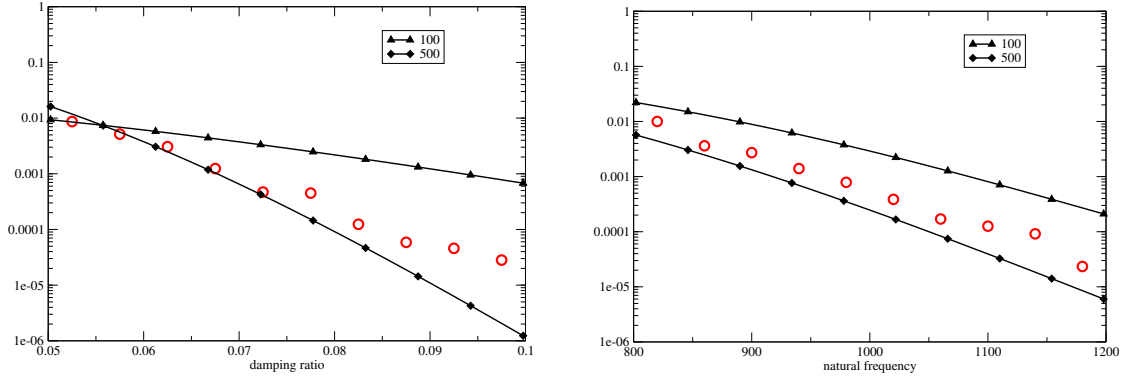


Figure 3: Comparison of reference solution (red circles) with estimates obtained using the probit regression schemes with 3 levels and 100 (triangle up) and 500 (diamond) training samples at each level. Left panel corresponds to $\omega_0 = 1000$ and right panel to $\xi = 0.07525$

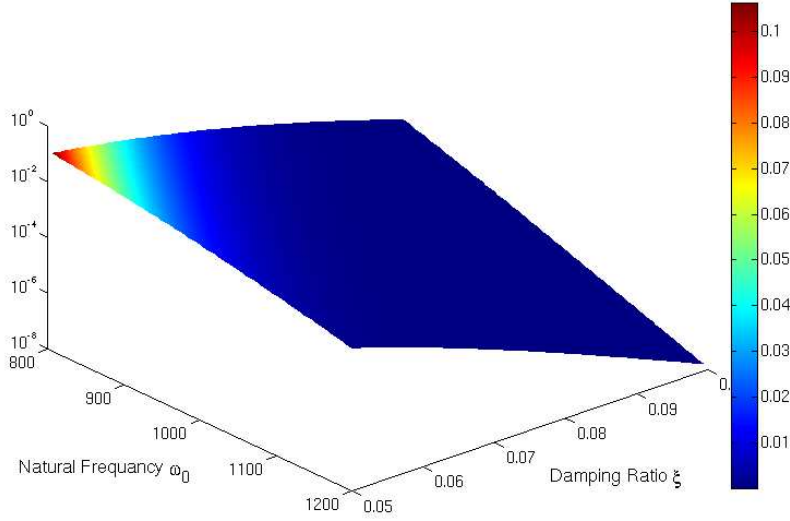


Figure 4: Estimates obtained using the probit regression schemes with 3 levels and 500 training samples at each level

measure of the effect of each design variable d_k on the probability of the event of interest. Figure 5 depicts the posterior distributions of the a_k that correspond to the natural frequency ω_0 and damping ratio ξ . In the iterative scheme that was applied, there were 3 levels which correspond to the events $|x(t)| > 0.0039$, $|x(t)| > 0.0048$ and $|x(t)| > 0.005$ (target). It can be clearly seen that the weight corresponding to ξ is larger (in absolute value) than the weight corresponding to ω_0^2 . This practically implies that a perturbation in the value of the former will cause a larger change in the probability of interest than an equal perturbation in the value of the latter. The iterative application provides also useful insight about the evolution of the relative importance of those variables at various response levels. Table 1 contains the posterior mean of the two model parameters a_k .

In the examples examined thus far, the dependence of the probability of interest is approximately log-linear w.r.t. \mathbf{d} . In addition, and since the dimension of the design space is fairly small (1 or 2), it is expected that the methods proposed in ([2] and [7]) would also perform well. In order to complicate matters a bit we consider a different

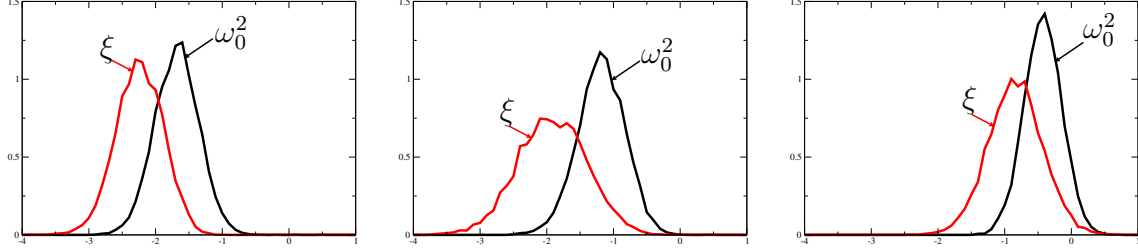


Figure 5: Comparison of reference solution (black circles) with estimates obtained using the probit regression schemes with 3 levels and 100 (triangle up) and 500 (diamond) training samples at each level. Left panel corresponds to $\omega_0 = 1000$ and right panel to $\xi = 0.07525$

| | a_{ω_0} | a_{ξ} |
|---------------------------------|----------------|-----------|
| Level 1 (k=2, $f_2 = -0.0039$) | -1.73 | -2.29 |
| Level 2 (k=1, $f_1 = -0.0048$) | -1.23 | -1.97 |
| Level 3 (k=0, $f_0 = -0.0050$) | -0.48 | -0.89 |

Table 1: Posterior means of probit regression weights

model for the random excitation Equation (26), namely a spectral representation of the form ([23]):

$$r(t) = \sqrt{2} \sum_{j=0}^J \sqrt{2S(\omega_k)\Delta\omega} \cos(\omega_k t + \phi_k) \quad (27)$$

where $\phi_k \sim i.i.d \ U[0, 2\pi]$. This corresponds to a zero-mean, Gaussian process with spectral density function $S(\omega)$. We selected $S(\omega)$ so that is peaked at two particular frequencies and as result the probability that the oscillator exceeds a given threshold would be peaked around those 2 frequencies due to resonance. The method in [7] would be unable to capture that effect due to the approximating form used (Equation (6)).

In particular, we assumed $J = 512$, $\Delta\omega = \frac{2\pi}{1024 \Delta T}$, and:

$$S(\omega_k) = \begin{cases} \frac{1}{4J\Delta\omega} + \frac{1}{40\Delta\omega} & \text{if } |\omega_k| = \omega_a \\ \frac{1}{4J\Delta\omega} + \frac{9}{40\Delta\omega} & \text{if } |\omega_k| = \omega_b \\ \frac{1}{4J\Delta\omega} & \text{if } |\omega_k| < J\Delta\omega \text{ and } \omega_k \neq \omega_a \text{ and } \omega_k \neq \omega_b \\ 0 & \text{elsewhere} \end{cases} \quad (28)$$

where $\omega_a = 49\Delta\omega = 30.07$ and $\omega_b = 74\Delta\omega = 45.4$ Figure 7 depicts the estimate obtained from the kernel model in the case that the damping ratio ξ is also a design variable. One can clearly distinguish the two peaks around ω_a and ω_b .

Figure 3 compares the results obtained using kernel and the traditional probit regression. The latter is unable to capture the nonlinear landscape of the probability of interest. The kernel probit model is able to adjust to the training data and when $N = 500$ samples are used at each step the differences with the reference solution are small.

4.2 Embankment Dam

We considered a cross section of an embankment dam which is depicted in Figure 8. Due to the nature of the material, one would expect substantial random variability in its properties. It is proposed to model the low-strain elastic and shear moduli as

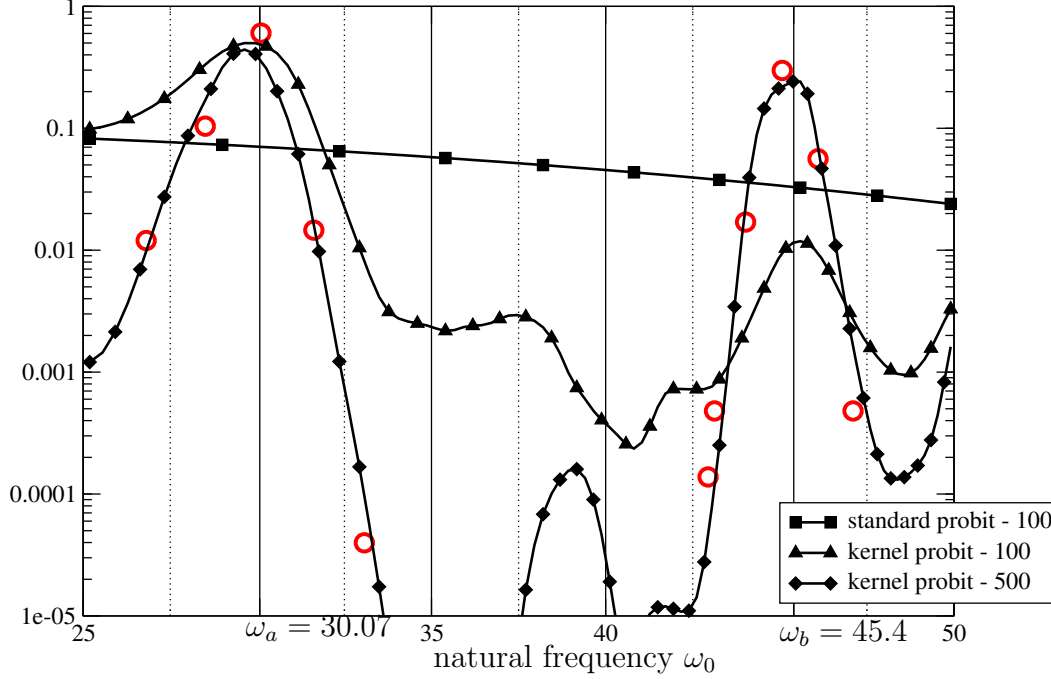


Figure 6: Estimates obtained using the probit regression schemes with 3 levels and 100 or 500 training samples at each level

non-homogeneous log-normal random fields $E(x, z)$ and $G(x, z)$:

$$E(x, z) = m_E(z) + \sigma_E(z) \frac{e^{Y(x, z)} - m_Y}{\sigma_Y}, \quad G(x, z) = m_G(z) + \sigma_G(z) \frac{e^{Y(x, z)} - m_Y}{\sigma_Y} \quad (29)$$

where $m_E(z) = E_0 - 0.4z$ (MPa), $m_G(z) = 12 - 0.1z$ (MPa) are the means, $\sigma_E(z) = \sigma_E$, $\sigma_G(z) = 5.0$ MPa are the standard deviations and $Y(x, z)$ is a homogeneous, zero mean, unit variance Gaussian random field with a given autocorrelation $R_Y(\Delta x, \Delta z) = e^{-\frac{|\Delta x|}{b_x} - \frac{|\Delta z|}{b_z}}$, $b_x = 10m$, $b_z = 3m$ ($m_Y = E[e^Y] = e^{0.5}$ and $\sigma_Y^2 = \text{Var}[e^Y] = e^2 - e$).

The cross section was discretized into 2160 triangular, 3-node, finite elements and the resulting system was solved under plane strain conditions in the linear elastic regime. Sample realizations of the material properties can be easily simulated by first simulating the underlying Gaussian field and mapping it according to Equation (29). The event of interest was considered to be failure with respect to the Coulomb criterion in at least one point in the analysis domain:

$$\tau \geq c + \sigma_n \tan \phi \quad (30)$$

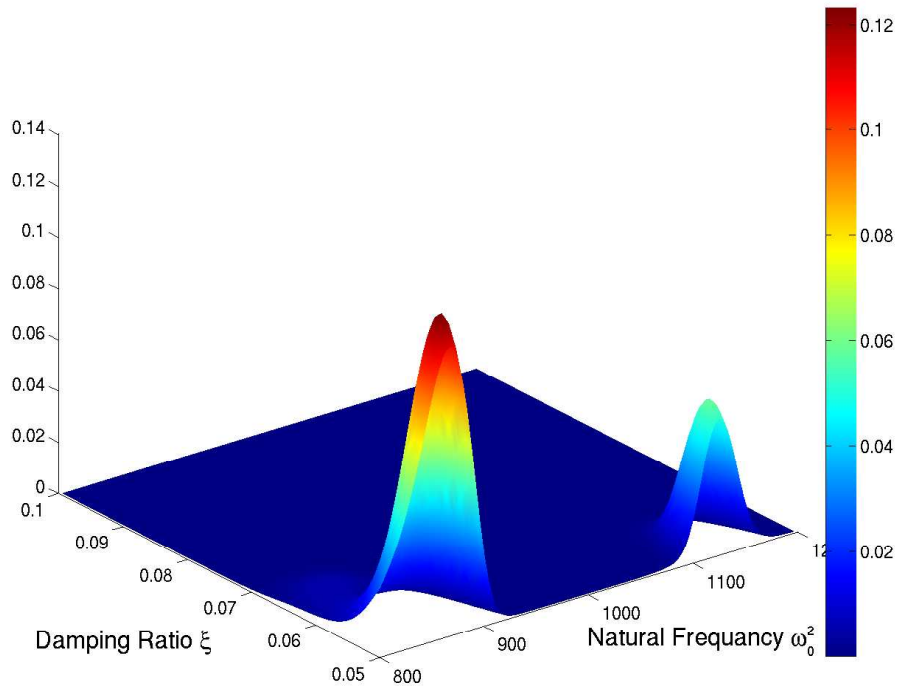


Figure 7: Estimates obtained using the kernel probit regression model with 3 levels and 1000 training samples at each level

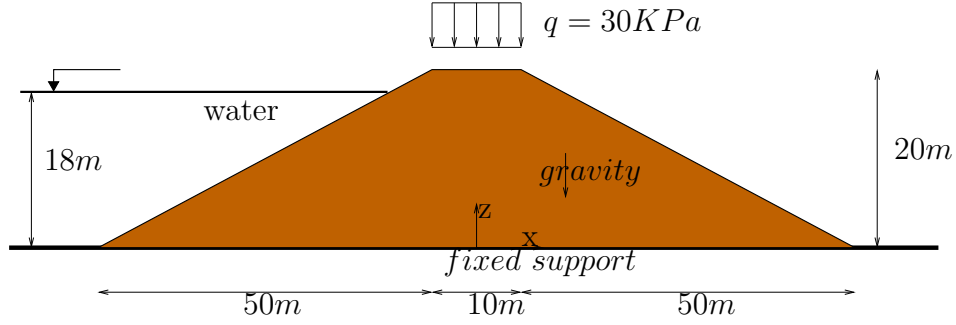


Figure 8: Schematic Illustration of

where τ is the shear stress, $c = 150 \text{ kPa}$ is the cohesive strength, σ_n is the normal stress (compressive stresses are considered positive) and $\phi = 40^\circ$ is the friction angle. Since triangular, 3-node elements are used the condition has to be checked once in every element.

The goal of this example is to examine the effect on the failure probability of different soil types or different soil processing techniques (i.e. compaction). Naturally there are several possibilities in describing the various design alternatives but for illustration purposes we consider here two design variables namely $E_0 \in [30, 40] \text{ (MPa)}$ and $\sigma_E \in [10, 15] \text{ MPa}$. For example a soil that has been subjected to compaction might still exhibit random variability in its properties but the mean values (reflected here with E_0) and the standard deviation of this variability (modeled by σ_E) will be higher and lower respectively. The results obtained can be also be combined with cost criteria in order to select the cheapest possible design satisfying a certain safety threshold or simultaneously optimizing with respect to cost and reliability. Figure 9 depicts the estimates obtained using the kernel probit regression model. A result that perhaps is unexpected and is revealed by the analysis is that the standard deviation σ_E has a more significant effect than the mean E_0 . This implies that, assuming all else being equal, a reduction in σ_E can cause a larger increase to the safety of the dam than an equivalent increase in E_0 .

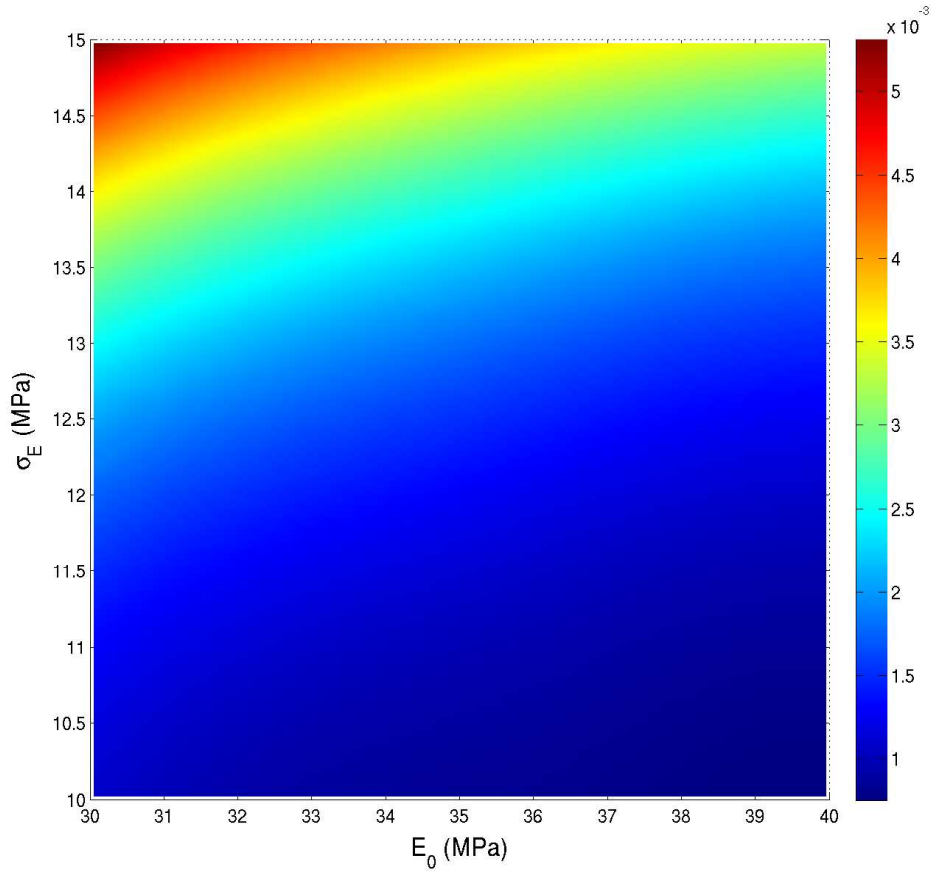


Figure 9: Estimates obtained using the probit regression schemes with 3 levels and 500 training samples at each level

4.3 Designing Random Microstructures - Fracture Modeling

The analysis of materials which exhibit very small length scales of heterogeneity has attracted considerable attention in recent years. This is because fine details in the microstructure can give rise to marked differences in the macroscale response. In reality, the majority of such materials exhibit randomness as local physical and mechanical properties fluctuate stochastically. In multiphase materials for example the distribution of the constituent phases in space does not follow a particular pattern and is characterized by disorder. It is therefore obvious that a probabilistic description is most appropriate and provides a sounder basis for their representation and the quantification of the reliability of the systems where these appear.

Even though during the fabrication process we can control certain statistics (i.e. volume fractions, spatial correlations), the resulting microstructure remains random. In complex, nonlinear deformation processes it is not a priori known how these features affect the response or rather the statistics of the response. Furthermore it is of interest to optimize the macroscale response with respect to those control/design variables.

To illustrate the usefulness of the proposed framework, we considered an 1D interface of unit length modeled by 1000 cohesive elements. These are line (or surface in 3D) elements which are located at the interfaces of adjacent bulk elements and govern their separation in accordance with a cohesive law. The concept of cohesive laws was pioneered by Dugdale ([13]) and Barenblatt ([4]) in order to model fracture processes and has been successfully used in a Finite Element setting by several researchers ([25, 5, 21]). According to these models, fracture is a gradual phenomenon in which separation takes place across an extended crack 'tip' or cohesive zone and is resisted by cohesive tractions. We assume herein a simple constitutive law relating interface traction-separation as seen in Figure 10. Under monotonic loading the normal interface traction decays as $T = T_c \left(1 - \frac{\delta}{\delta_c}\right)$ for $\delta \leq \delta_c$ and $T = 0$ for $\delta > \delta_c$. The fracture energy G_c is given by $G_c = T_c \delta_c / 2$.

At the microstructural level, the cohesive properties exhibit random variability.

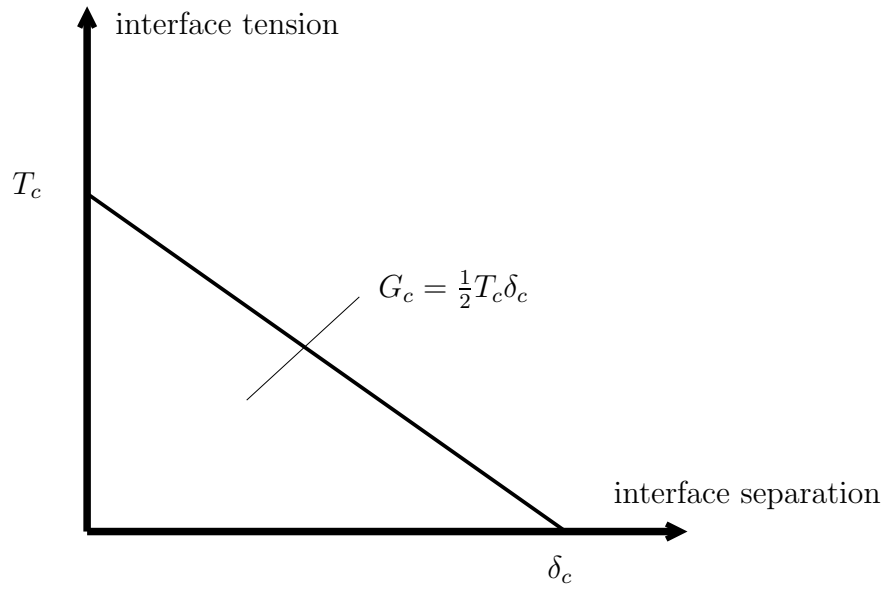


Figure 10: Cohesive Law: T_c denotes ultimate interfacial tension (when the stress reaches T_c the cohesive element is activated), δ_c denotes the ultimate separation interface (when the separation reaches δ_c the interface tension becomes zero) and G_c denotes the fracture energy which is equal to the area under the tension-separation curve.

Consider for example a carbon fiber-reinforced composite which are currently used in aerospace applications. It is natural to expect that T_c and G_c will change from matrix to fiber which in turn are randomly distributed within the composite. We adopt the following simple model based on a uniformly distributed random field:

$$\begin{aligned} T_c(z) &= T_1 + \Delta T U(z) \\ G_c(z) &= G_1 + \Delta G U(z) \quad z \in [0, 1] \end{aligned} \quad (31)$$

where:

$$U(z) = \Phi(h(z)) \quad (32)$$

and $h(z)$ is a zero-mean, unit variance Gaussian process with autocorrelation $R_h(\Delta z) = E[h(z)h(z + \Delta z)] = \exp\{-\frac{|\Delta z|}{z_0}\}$. The parameter z_0 controls the length scale of heterogeneity and it was taken equal to 0.1

The macroscale response will also be random. Figure 11 depicts five realizations of the possible traction-separation histories for the whole interface where the large variability in the maximum traction as well as fracture energy (area under the curve) can be easily observed. Naturally design criteria are formulated with respect to the macroscale response, i.e. the fracture energy associated with the whole interface. It is of interest therefore to examine how the statistics of the response depend on design variables associated with the microstructure. In the present example the role of the design variables is played by the parameters T_1 , ΔT , G_1 , ΔG (Equation (31)) which control the mean and variance of the cohesive strength and fracture energy at the microstructural level. Figures 12 and 13 depict how the probability that the macro fracture energy $\hat{G} \geq G_0 = 0.7$ depends on these design variables which lie in the intervals $T_1 \in [1.0, 1.5]$, $\Delta T \in [0.1, 0.5]$, $G_1 \in [1.0, 1.5]$, $\Delta G \in [0.1, 0.5]$. It was found that the response statistic was more sensitive to the design variables associated with cohesive strength T_1 and ΔT rather than fracture energy G_1 and δG at the microstructural level. Furthermore, a result that is perhaps counterintuitive, is that increasing ΔT (i.e. the variability of the cohesive strength at the microstructural level), increases the

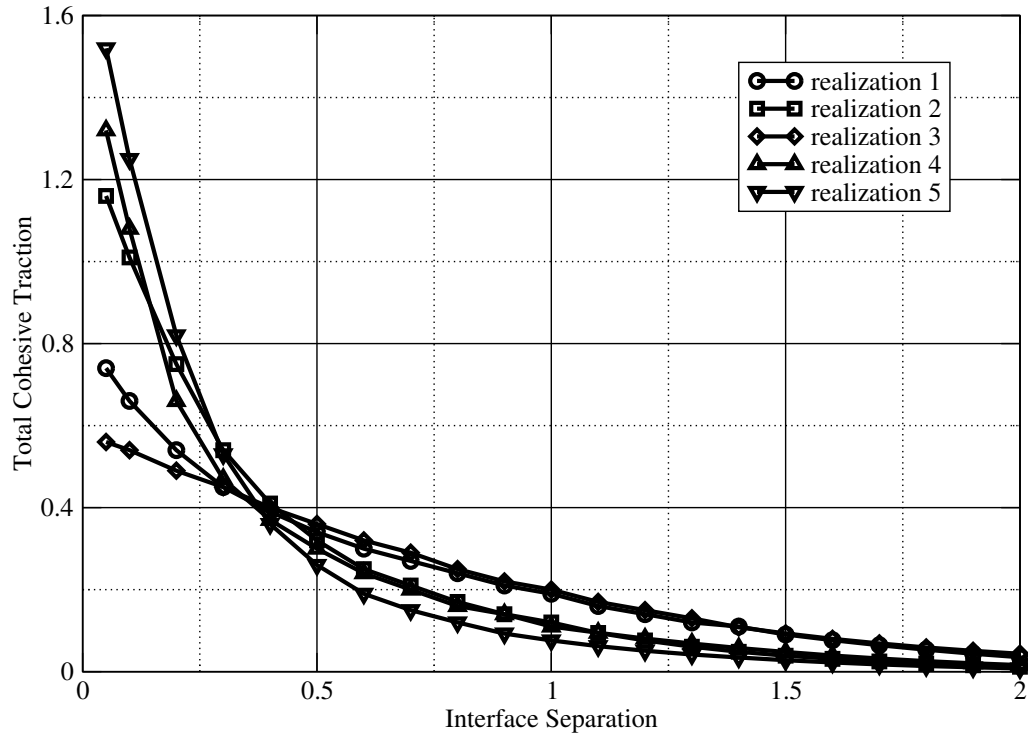


Figure 11: Sample realizations of the total traction-separation along the cohesive interface when the cohesive strength and cohesive energy at the microscale varies as in Equation (31)

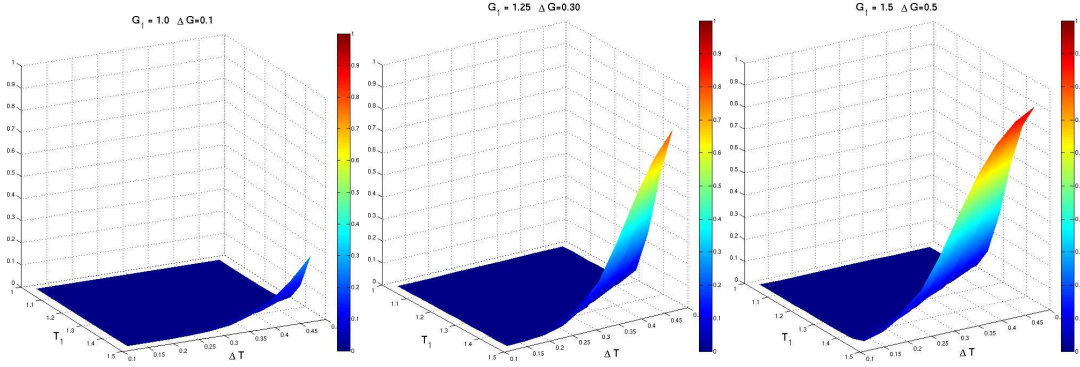


Figure 12: Estimates of the probability of interest for three sets of values of G_1 and ΔG .

probability that the fracture energy at the macroscale \hat{G} exceeds the specified threshold G_0 .

4.4 Policy Decisions for Energy Systems

The mathematical modeling of energy systems at local or national levels has experienced significant advances in recent years. Several models have been developed that can, to a certain extent, predict the evolution of such systems but also provide optimal strategies. Nevertheless the predictive ability of these models is hampered by significant uncertainties in several parameters. These can have a tremendous effect in determining optimal future policies. A policy that is favorable, or even optimal for current oil prices, will generally not be so if the latter drops or increases significantly.

The importance of these factors has become more pronounced nowadays as, apart from economical implications, we also need to consider environmental consequences (e.g. CO_2 emissions). Future policy decisions should be made in the presence of uncertainty and any optimization should utilize as objectives not the performance metrics (i.e. overall cost) which are themselves random but rather statistics of the latter (i.e. the probability that the cost stays below a certain threshold or probability CO_2 emissions are reduced by a certain amount given a cost constraint). Hence there is a need for a flexible and efficient computational environment that can utilize existing

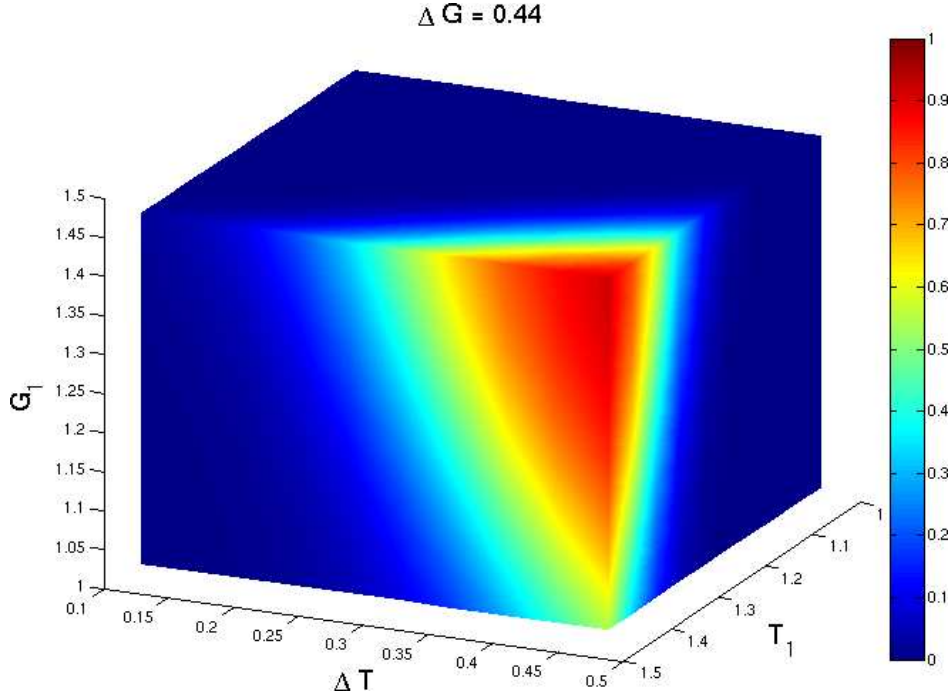


Figure 13: Estimates of the probability of interest for $\Delta G = 0.44$.

deterministic models in order to quantify the effect of uncertainties and the sensitivity to design/policy variables.

In this example we make use of MARKAL (<http://www.etsap.org/markal/main.html>), a deterministic solver that models the evolution over a period of usually 40 to 50 years of a specific energy system at the national, regional, state or province, or community level. This plays the role of the black-box solver in our proposed framework. Naturally the input of the model consists of an extremely large number of parameters. Several of those exhibit significant uncertainties and must therefore be modeled in a probabilistic framework. Other input parameters correspond to design or policy variables which can be adjusted in order to achieve a desirable outcome/performance. In this example we considered 2 random variables corresponding to the future evolution of oil and natural gas prices and four design/policy variables that pertain to CO_2 emissions threshold, Nuclear Capacity Expansion, Sequestration Capacity Expansion and Renewable Capacity Expansion. Figure 14 depicts the probability that the overall cost will not exceed by more than 1% the current GDP for various CO_2 reduction levels (with respect to

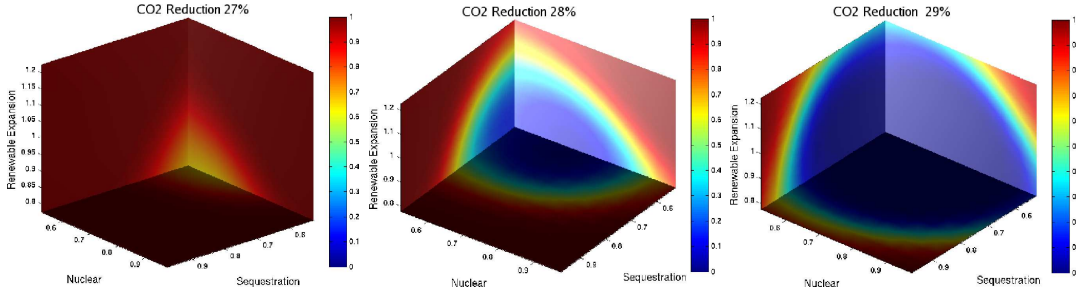


Figure 14: Estimates of the probability of interest for three CO_2 reduction levels

current emissions) as a function of the three remaining policy variables.

5 CONCLUSIONS

In the majority of engineering systems, the design requirements must be addressed by accounting for unavoidable uncertainties. The proposed computational framework can interact in a non-intrusive manner with any deterministic solver (e.g. finite element codes) in order to quantify response uncertainties. The sequential importance sampling scheme maximizes the amount of information extracted about the system with a given number of calls to the deterministic black box. The algorithm can perform equally well for very large numbers of random variables. Probabilistic classifiers (e.g. probit regression) can provide accurate estimates of the output probabilities of interest as a function of the design variables. Their performance can be greatly improved by combining independent models trained on a subset of the data. Kernel-based versions can also improve the flexibility of the model. The two steps in the proposed algorithm framework (stochastic sampling & statistical learning) are now independent. The overall performance could however be greatly improved if they are employed in conjunction and in an adaptive manner.

References

- [1] J. Albert and S. Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88:669–679, 1993.
- [2] S.K. Au. Reliability-based design sensitivity by efficient simulation. *Computers & Structures*, 83(14):1048–1061, 2005.
- [3] S.K. Au and J. Beck. Estimation of small failure probabilities in high dimensions by subset simulation. *Probabilistic Engineering Mechanics*, 16:263–277, 2001.
- [4] G.I. Barenblatt. The mathematical theory of equilibrium of cracks in brittle fracture. *Advances in Applied Mechanics*, 7:55, 1962.
- [5] G.T. Camacho and M. Ortiz. Computational modeling of impact damage in brittle materials. *International Journal of Solids and Structures*, 33(20):2899, 1996.
- [6] O. Cappé, A. Guillin, J.M. Marin, and C. P. Robert. Population Monte Carlo. *Journal of Computational and Graphical Statistics*, 13:907–930, 2004.
- [7] J. Ching and Y.-H. Hsieh. Local estimation of failure probability function and its confidence interval with maximum entropy principle. *Probabilistic Engineering Mechanics*, 22(1):39–49, 2007.
- [8] N. Chopin. A sequential particle filter method for static models. *Biometrika*, 89:539–552, 2002.
- [9] P. Del Moral, A. Doucet, and A. Jasrau. Sequential Monte Carlo Samplers. *Journal of the Royal Statistical Society B*, 68(3):411–436, 2006.
- [10] A.J. Dobson. *An Introduction to Generalized Linear Models*. Chapman & Hall, 2nd edition, 2002.

- [11] R. Douc and O. Cappé. Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, 2005.
- [12] A. Doucet, J. F. G. de Freitas, and N. J. Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer, New York, 2001.
- [13] D.S. Dugdale. Yielding of steel sheets containing clits. *Journal of the Mechanics and Physics of Solids*, 8:100, 1960.
- [14] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin. *Bayesian Data Analysis*. Chapman & Hall/CRC, 2nd edition, 2003.
- [15] W. E. Griffiths, R.C. Hill, and P.J. Pope. Small samples properties of probit model estimators. *Journal of the American Statistical Association*, 82:929–937, 1987.
- [16] C.C. Holmes and K. Held. Bayesian auxiliary variable models for binary and polychotomous regression. *to appear in Bayesian Analysis*, 2007.
- [17] E. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106:620–630, 1957.
- [18] J.S. Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Series in Statistics. Springer, 2001.
- [19] S.N. MacEachern, M. Clyde, and J.S. Liu. Sequential importance sampling for nonparametric bayes models: The next generation. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 27(2):251–267, 1998.
- [20] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 1(1):125–139, 2001.
- [21] M. Ortiz and A. Pandolfi. Finite-deformation irreversible cohesive elements for three-dimensional crack propagation analysis. *International Journal for Numerical Methods in Engineering*, 44:1267, 1999.

- [22] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer, 2004.
- [23] M. Shinozuka and G. Deodatis. Simulation of multi-dimensional stochastic processes by spectral representation. *ASME Applied Mechanics Reviews*, 49(1):29–53, 1996.
- [24] G. Wahba. Support vector machines, reproducing kernel hilbert spaces and the randomized gacv. In Burges Schölkopf and Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, pages 69–88.
- [25] X.P. Xu and A. Needleman. Numerical simulations of fast crack growth in brittle solids. *Journal of the Mechanics and Physics of Solids*, 42:1397, 1994.